Language prediction model

Ideas for improvements

This documents contains a couple of exercises which should either increase the model's accuracy or make it more robust. All these exercises are optional and have no dead-line.

Feature drop-out

The goal of this exercise is to implement drop-out over n-gram features. The current model easily over-fits by capturing (potentially accidental) patterns occurring between different n-grams in the same input name. To alleviate this:

- Add a parameter of the `LangRec` class which specifies the probability of discarding each (n-gram) feature in a given input name.
- Modify the `forward` method so that each feature of the given name is discarded with the pre-specified probability.

Warning: drop-out should only be applied during training!

Case-insensitive model

The current model can assign different scores to names which differ in case, e.g. *adam*, *Adam*, and *ADAM*. Propose and implement a pre-processing method which guarantees that all the three versions of *Adam* get the same scores.

Balanced model

The distribution of names in our dataset is not balanced (e.g., most of them are Russian). Propose a modification of the objective loss function which will treat all the languages as equally important, regardless of the number of the corresponding examples in the dataset.

Hint: Have a look at the PyTorch documentation of the cross-entropy loss.

Disclosure: didn't try this one yet!

Name-length bias

As pointed out before, due to the use of the `sum` variant of CBOW, our model can learn to (indirectly) rely on the length of the input name when making predictions. Propose a modification of the model which, by design, is less length-sensitive.

Hint: the idea was already mentioned during one of the practical sessions.